

Appl. No. : 09/242,383
Filed : February 12, 1999

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application. The listing of claims presents each claim with its respective status shown in parentheses.

Listing of Claims

Claims 1-74 (Canceled).

Claim 75 (Previously presented): A computing device programmed to manipulate or access objects of the string class using an object oriented operating system, wherein the objects of the string class are derived from a single base class and the operating system handles all such objects of the string class according to one or more of the following requirements:

- (a) objects of the string class for literal text are handled as belonging to a class in which a pointer points to the memory location where the text string is stored;
- (b) objects of the string class for length limited text are handled as belonging to a class in which a buffer stores text of a predetermined length requiring a limited subset of available memory management functions; and
- (c) objects of the string class using heap memory are handled as belonging to a class requiring the full set of available memory management functions.

Claim 76 (Previously presented): The computing device of Claim 75, further comprising a program which interfaces with the operating system and which also handles objects according to one or more of the requirements.

Claim 77 (Previously presented): The computing device of Claim 75, wherein objects satisfying one or more of the requirements are flat structures.

Claim 78 (Previously presented): The computing device of Claim 75, wherein objects of the string class for length limited text are stored in particular memory locations at run time which are not part of the heap memory.

Claim 79 (Previously presented): The computing device of Claim 75, wherein the objects are polymorphic.

Appl. No. : 09/242,383
Filed : February 12, 1999

Claim 80 (Previously presented): The device of Claim 79, wherein polymorphism is achieved by providing a data field for each object which identifies its class, with a different action being associated with different data field values.

Claim 81 (Previously presented): The computing device of Claim 80, wherein the data field is a part of the representation of another data item within a machine word.

Claim 82 (Previously presented): The computing device of Claim 81, wherein the same source code is used, irrespective of the character code system and character code width being used, by using aliases for class names that are character code independent.

Claim 83 (Previously presented): The computing device of Claim 82, wherein the source code using text strings is written in a manner independent of the strings' actual ASCII or Unicode implementation by using a system of aliases for class names.

Claim 84 (Previously presented): The computing device of Claim 75, wherein objects have information on the length of the data they contain and hence have no '0' terminator.

Claim 85 (Previously presented): A method of allowing objects of the string class to be manipulated or accessed by a program using an object oriented operating system, wherein the program handles all such objects according to one or more of the following requirements:

- (a) objects of the string class for literal text are handled as belonging to a class in which a pointer points to the memory location where the text string is stored;
- (b) objects of the string class for length limited text are handled as belonging to a class in which a buffer stores text of a predetermined length requiring a limited subset of available memory management functions; and
- (c) objects of the string class using heap memory are handled as belonging to a class requiring the full set of available memory management functions.

Claim 86 (Previously presented): The method of Claim 85, being performed by an operating system.

Claim 87 (Currently Amended): The method of Claim 85, being performed by a program which interfaces with an operating system which itself also performs the method of Claim ~~86~~ 85.

Claim 88 (Previously presented): The method of Claim 85, wherein objects satisfying one or more of the requirements are flat structures.

Appl. No. : 09/242,383
Filed : February 12, 1999

Claim 89 (Previously presented): The method of Claim 85, wherein objects of the string class for length limited text are stored in particular memory locations at run time which are not part of the heap memory.

Claim 90 (Previously presented): The method of Claim 85, wherein the objects are polymorphic.

Claim 91 (Previously presented): The method of Claim 90, wherein polymorphism is achieved by providing a data field for each object which identifies its class, with a different action being associated with different data field values.

Claim 92 (Previously presented): The method of Claim 91, wherein the data field is a part of the representation of another data item within a machine word.

Claim 93 (Previously presented): The method of Claim 92, wherein the same source code is used, irrespective of the character code system and character code width being used, by using aliases for class names that are character code independent.

Claim 94 (Previously presented): The method of Claim 93, wherein the source code using text strings is written in a manner independent of the strings' actual ASCII or Unicode implementation by using a system of aliases for class names.

Claim 95 (Previously presented): The method of Claim 85, wherein objects have information on the length of the data they contain and hence have no '0' terminator.

Claim 96 (Previously presented): Computer software which allows objects of the string class to be manipulated or accessed by a program using an object oriented operating system, wherein the program handles all such objects according to one or more of the following requirements:

- (a) objects of the string class for literal text are handled as belonging to a class in which a pointer points to the memory location where the text string is stored;
- (b) objects of the string class for length limited text are handled as belonging to a class in which a buffer stores text of a predetermined length requiring a limited subset of available memory management functions; and
- (c) objects of the string class using heap memory are handled as belonging to a class requiring the full set of available memory management functions.

Appl. No. : 09/242,383
Filed : February 12, 1999

Claim 97 (Previously presented): The computer software of Claim 96, being an object oriented operating system.

Claim 98 (Currently amended): The computer software of Claim ~~96~~ 97, ~~being~~ further comprising a program which is capable of interfacing with the object oriented operating system ~~of Claim 97~~.

Claim 99 (Previously presented): The computer software of Claim 96, wherein in which objects satisfying one or more of the requirements are flat structures.

Claim 100 (Previously presented): The computer software of Claim 96, wherein objects of the string class for length limited text are stored in particular memory locations at run time which are not part of the heap memory.

Claim 101 (Previously presented): The computer software of Claim 96, wherein the objects are polymorphic.

Claim 102 (Previously presented): The computer software of Claim 101, wherein polymorphism is achieved by providing a data field for each object which identifies its class, with a different action being associated with different data field values.

Claim 103 (Previously presented): The computer software of Claim 102, wherein the data field is a part of the representation of another data item within a machine word.

Claim 104 (Previously presented): The computer software of Claim 103, wherein the same source code is used, irrespective of the character code system and character code width being used, by using aliases for class names that are character code independent.

Claim 105 (Previously presented): The computer software of Claim 104, wherein the source code using text strings is written in a manner independent of the strings' actual ASCII or Unicode implementation by using a system of aliases for class names.

Claim 106 (Previously presented): The computer software of Claim 96, wherein objects have information on the length of the data they contain and hence have no '0' terminator.

Appl. No. : 09/242,383
Filed : February 12, 1999

SUMMARY OF INTERVIEWS

The undersigned conducted telephone interviews with the Examiner on May 26, 2004 and June 16, 2004. The following summarizes these interviews.

Identification of Claims Discussed

Claims 39-106 were discussed.

Identification of Prior Art Discussed

The following references were discussed: (1) Henricson et al., "Programming in C++ Rules and Recommendations" and (2) Cowlshaw, "The REXX Language A Practical Approach to Programming."

Proposed Amendments

On June 16, 2004, the Applicant proposed canceling Claims 39-74 without prejudice or disclaimer. The Applicant informed the Examiner that Claims 39-74 would likely be filed at a later date in a continuation application.

Principal Arguments and Other Matters

On May 26, 2004, the Applicant asked the Examiner to explain the rejection of claims 39-106 under 35 U.S.C. § 103(a). The language of Claim 40 was discussed in view of Henricson and Cowlshaw. No agreement was reached.

On June 16, 2004, the Applicant proposed canceling claims 39-74 without prejudice so as to focus arguments on Claims 75-106 in the present Amendment. The Examiner indicated that he would reconsider Claims 75-106 and would call the undersigned for further discussion, if necessary, after considering the present Amendment.

Results of Interviews

As a result of the interviews, the Applicant has proposed canceling Claims 39-74 herein and has provided arguments as to the patentability of Claims 75-106.